Supplement to
Outlier Detection for Multi-Network Data

Pritam Dey, Zhengwu Zhang, and David B. Dunson

This document contain additional material including mathematical results, algorithms and other details to support the paper 'Outlier Detection for Multi-Network Data.' All mathematical notations introduced in the main paper are also valid for this document.

S1 Mathematical Proofs and Results

Lemma S1.1. *(i)* ***Inverse of block matrix:*** *If an $n \times n$ symmetric matrix M is partitioned into four blocks as $M = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$, where A and C are invertible square matrices of arbitrary size and the Schur complement of C in M $(A - BC^{-1}B^T)$ is invertible, then the inverse of M can be expressed as:*

$$M^{-1} = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}^{-1} = \begin{bmatrix} (A - BC^{-1}B^T)^{-1} & -(A - BC^{-1}B^T)^{-1}BC^{-1} \\ -C^{-1}B^T(A - BC^{-1}B^T)^{-1} & C^{-1} + C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1} \end{bmatrix} \tag{S1}$$

(ii) If $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = M^{-1} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ then the following formulas hold:

$$x_1 = (A - BC^{-1}B^T)^{-1}(v_1 - BC^{-1}v_2) \tag{S2}$$
$$x_2 = C^{-1}(v_2 - B^T x_1) \tag{S3}$$

Proof. Part (i) is a standard result which can be verified by matrix multiplication of M and M^{-1} in the given forms. Part (ii) follows easily from part (i) by explicit multiplication and simple algebraic manipulations. $\square$

S1.1 MM algorithm for estimation of $(Z, \beta_1, \beta_2, ..., \beta_N)$

Concatenating $(Z, \beta_1, \beta_2, ..., \beta_N)$ into a single parameter vector θ in that order, allows us to calculate the gradient vector and Hessian matrix:

$$\nabla \mathcal{L}(Z, \{\beta_j\}_{j=1}^N) = \frac{1}{N} \begin{pmatrix} \sum_{i=1}^N (\boldsymbol{a}_i - \boldsymbol{\pi}_i) \\ X^T(\boldsymbol{a}_1 - \boldsymbol{\pi}_1) \\ X^T(\boldsymbol{a}_2 - \boldsymbol{\pi}_2) \\ \vdots \\ X^T(\boldsymbol{a}_N - \boldsymbol{\pi}_N) \end{pmatrix} - \begin{pmatrix} \lambda Z \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{S4}$$

$$\mathcal{H}(Z, \{\beta_j\}_{j=1}^N) = -\frac{1}{N} \left[\begin{array}{c|ccc} \sum_{i=1}^N W_i & W_1^T X & \dots & W_N^T X \\ \hline X^T W_1 & X^T W_1 X & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ X^T W_N & 0 & \dots & X^T W_N X \end{array}\right] - \lambda \left[\begin{array}{c|ccc} I & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{array}\right] \tag{S5}$$

The first matrix in the equation (S5) can be expressed as $\tilde{X}^T W \tilde{X}$ where W is a diagonal matrix consisting of the matrices $W_1, W_2, ..., W_N$ stacked diagonally into a single matrix and

$$\tilde{X} = \left[\begin{array}{c|cccc} I & X & 0 & \dots & 0 \\ I & 0 & X & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & 0 & 0 & \dots & X \end{array}\right] \tag{S6}$$

Further we note that every entry in the diagonal matrix W is bounded above by $\frac{1}{4}$. So, the difference $\frac{1}{4}\tilde{X}^T\tilde{X} - \tilde{X}^T W \tilde{X}$ is non-negative definite which gives us the following minorizing function for MM algorithm,

$$g\left(\theta \mid \theta^{(k)}\right) = \mathcal{L}\left(\theta^{(k)}\right) + \nabla\mathcal{L}\left(\theta^{(k)}\right)^T \left(\theta - \theta^{(k)}\right) + \frac{1}{2}\left(\theta - \theta^{(k)}\right)^T M \left(\theta - \theta^{(k)}\right) \tag{S7}$$

where M is obtained by replacing $\frac{1}{4}\tilde{X}^T\tilde{X}$ in place of $\tilde{X}^T W \tilde{X}$ in the Hessian matrix. So,

$$M = -\frac{1}{4N} \left[\begin{array}{c|cccc} (4\lambda + 1)NI & X & X & \dots & X \\ \hline X^T & X^T X & 0 & \dots & 0 \\ X^T & 0 & X^T X & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X^T & 0 & 0 & \dots & X^T X \end{array}\right] \tag{S8}$$

The maximizer of $g\left(\theta \mid \theta^{(k)}\right)$ is then $\theta^{(k+1)} = \theta^{(k)} - M^{-1}\nabla\mathcal{L}\left(\theta^{(k)}\right)$ which means that using part (ii) of the lemma, the updates of Z and β_i are:

$$Z^{(k+1)} = Z^{(k)} + 4\left[4\lambda I + I - X(X^T X)^{-1}X^T\right]^{-1} \left[-\lambda Z^{(k)} + \left(I - X(X^T X)^{-1}X^T\right) \frac{\sum_{i=1}^{N}\left(\boldsymbol{a}_i - \boldsymbol{\pi}_i^{(k)}\right)}{N}\right] \tag{S9}$$

$$\beta_i^{(k+1)} = \beta_i^{(k)} + 4(X^T X)^{-1}X^T\left[\left(\boldsymbol{a}_i - \boldsymbol{\pi}_i^{(k)}\right) - N\left(Z^{(k+1)} - Z^{(k)}\right)\right] \tag{S10}$$

As we are not updating the whole vector at once, we save a lot of computation with large matrices. The β_i's are not very large in dimension in general, so the only computation with large matrices that has to be done in every iteration are the matrix multiplications in the update of Z.

Algorithm 1: MM Algorithm for estimation of $(Z, \beta_1, \beta_2, \dots, \beta_N)$

Data: Binary adjacency matrices: $A_1, A_2, \dots, A_N$

Input: Hemisphere and lobe memberships for every ROI in the atlas, λ and a tolerance value for the stopping condition.

Output: Estimated parameter vectors $(\widehat{Z}, \widehat{\beta}_1, \widehat{\beta}_2, \dots, \widehat{\beta}_N)$

Vectorize the adjacency matrices by taking only the elements below the diagonal to get vectors $\{\boldsymbol{a}_i\}_{i=1}^N$.

Form the X matrix using the hemisphere and lobe relationships of every pair of ROIs.

Calculate the matrices $Q = I - X(X^T X)^{-1}X^T$, $R = 4(Q + 4\lambda I)^{-1}$ and $S = 4(X^T X)^{-1}X^T$.

Initialize the parameters $k = 0$, $Z = Z^{(0)}$, $\beta_i = \beta_i^{(0)}$ and evaluate the objective function at the initial values.

while *not stopping condition* **do**

  Calculate $\left\{\left(\boldsymbol{a}_i - \boldsymbol{\pi}_i^{(k)}\right) = \boldsymbol{a}_i - logit\left(Z^{(k)} + X\beta_i^{(k)}\right)\right\}_{i=1}^N$ and $\mu^{(k)} = \frac{\sum_{i=1}^N\left(\boldsymbol{a}_i - \boldsymbol{\pi}_i^{(k)}\right)}{N}$

  Update $\boldsymbol{Z}$: $Z^{(k+1)} = Z^{(k)} + R\left[-\lambda Z^{(k)} + Q\mu^{(k)}\right]$

  Update $\boldsymbol{\beta_i}$: $\beta_i^{(k+1)} = \beta_i^{(k)} + S\left[\left(\boldsymbol{a}_i - \boldsymbol{\pi}_i^{(k)}\right) + N\left(Z^{(k+1)} - Z^{(k)}\right)\right]\ \forall i$

  Calculate objective function at $Z^{(k+1)}$ and $\left\{\beta_i^{(k+1)}\right\}_{i=1}^N$

  Set k=k+1

return $\widehat{Z} = Z^{(k-1)}$ and $\left\{\widehat{\beta}_i = \beta_i^{(k-1)}\right\}_{i=1}^N$

S1.2 Proof of Proposition 1

The objective function to minimize to estimate the model parameters after excluding the i^{th} unit is given by $\mathcal{L}(\widehat{Z}, \{\widehat{\beta}_j\}_{j\neq i}) = \frac{1}{N-1}\sum_{j\neq i}\sum_{l=1}^{L}\left[a_{jl}\widehat{\eta_{jl}} - log(1+e^{\widehat{\eta_{jl}}})\right] - \frac{\lambda}{2}\left\|\widehat{Z}\right\|_2^2$. Using the lemma S1.1, and the gradient and hessian given in S4 and S5 the first Newton Raphson update in Z starting from $\widehat{Z}$ is:

$$Z_{-i}^{(1)} = \widehat{Z} + \left[\lambda I + \frac{\sum_{j\neq i} W_j}{N-1} - \frac{\sum_{j\neq i} B_j^T Q_j^{-1} B_j}{N-1}\right]^{-1} \left[\frac{\sum_{j\neq i}\left(I - B_j^T Q_j^{-1} X^T\right)(\boldsymbol{a}_j - \widehat{\boldsymbol{\pi}}_j)}{(N-1)} - \lambda\widehat{Z}\right] \tag{S11}$$

But since $(\widehat{Z}, \widehat{\beta}_1, ..., \widehat{\beta}_N)$ is the pMLE, $\nabla_Z \mathcal{L}(\widehat{Z}, \{\widehat{\beta}_j\}_{j=1}^N) = \frac{1}{N}\sum_{i=1}^N(\boldsymbol{a}_i - \widehat{\boldsymbol{\pi}}_i) - \lambda\widehat{Z} = \mathbf{0}$ and $\nabla_{\beta_j}\mathcal{L}(\widehat{Z}, \{\widehat{\beta}_j\}_{j=1}^N) = \frac{1}{N}X^T(\boldsymbol{a}_j - \widehat{\boldsymbol{\pi}}_j) = \mathbf{0}$. So,

$$\begin{aligned}\frac{1}{N-1}\sum_{j\neq i}\left(I - B_j^T Q_j^{-1} X^T\right)(\boldsymbol{a}_j - \widehat{\boldsymbol{\pi}}_j) - \lambda\widehat{Z} &= \frac{1}{N-1}\sum_{j\neq i}(\boldsymbol{a}_j - \widehat{\boldsymbol{\pi}}_j) - \lambda\widehat{Z} \\ &= -\frac{1}{N-1}\left[(\boldsymbol{a}_i - \widehat{\boldsymbol{\pi}}_i) - \lambda\widehat{Z}\right]\end{aligned} \tag{S12}$$

So, putting all this together, we get

$$\begin{aligned}Z_{-i}^{(1)} - \widehat{Z} &= -\frac{1}{N-1}\left[\lambda I + \frac{\sum_{j\neq i} W_j}{N-1} - \frac{\sum_{j\neq i} B_j^T Q_j^{-1} B_j}{N-1}\right]^{-1}\left[(\boldsymbol{a}_i - \widehat{\boldsymbol{\pi}}_i) - \lambda\widehat{Z}\right] \\ &= -\left[\lambda(N-1)I + \sum_{j\neq i} W_j - \sum_{j\neq i} B_j^T Q_j^{-1} B_j\right]^{-1}\left[(\boldsymbol{a}_i - \widehat{\boldsymbol{\pi}}_i) - \lambda\widehat{Z}\right]\end{aligned} \tag{S13}$$

□

S2 Thresholding influence measures for outlier detection

The basic idea is that most of the subjects have very low values for these influence measures. Only a handful will possibly have high values of these measures.

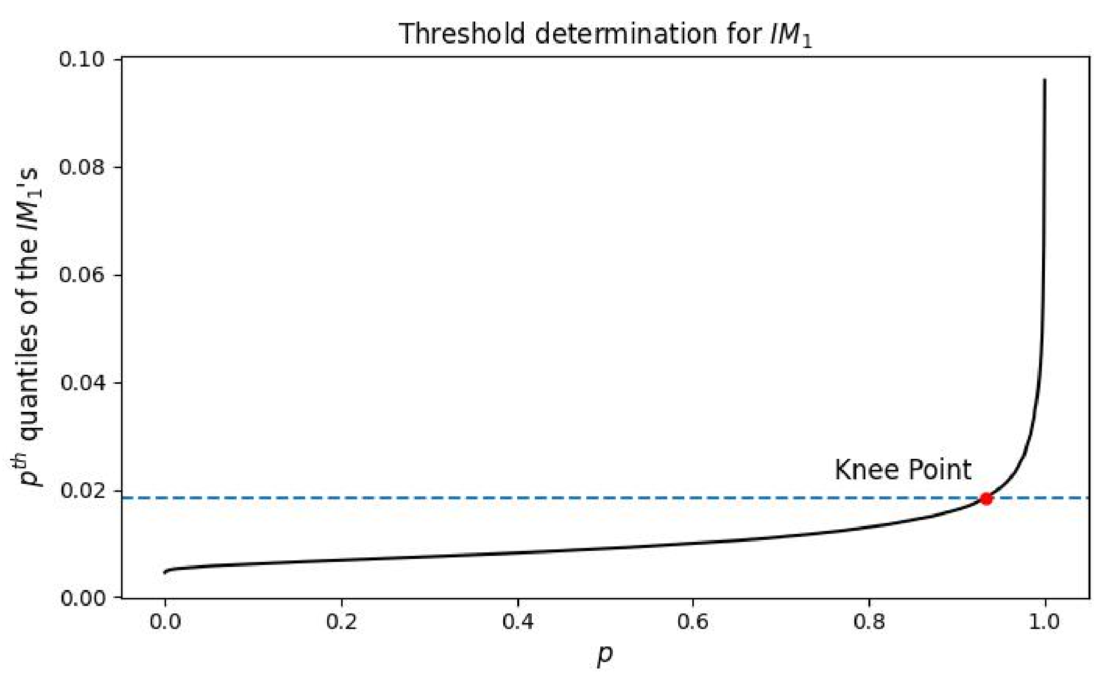

Figure S1: A plot of the quantiles of $IM_2(i)$. Here we plot the p^{th} quantile of the $IM_2(i)$ values for $p \in [0, 1]$. As mentioned in section 2 of the paper, the graph remains relatively flat for most of the lower quantiles and then starts to sharply increase as we get to the higher quantiles. The 'elbow' of this plot, i.e. the point where the graph starts increasing sharply can be used as a threshold. This point is shown in red in the graph. It is found using the 'kneedle' algorithm (Satopaa *et al.*, 2011).

S3 Additional description of the regions of interest for the circular plots

The circular plots in the main document, an example of which is given here have several dots arranged around a circular region. Each dot represents a region of interest. These dots are colour-coded to represent which lobe and hemisphere they come from.

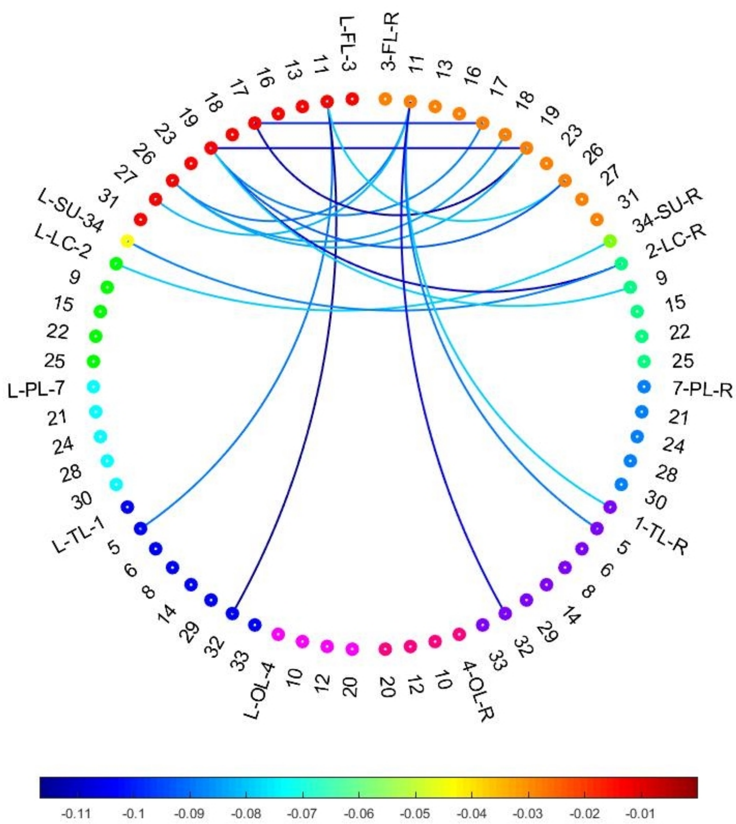

Figure S2: An example of the circular plots in the application section in the main document.

Each ROI is indicated by a number which is its position in the Desikan atlas. For one of the ROI's in each lobe and hemisphere combination, in addition to the numbers there is also a text in the figure of the form "X-YY". Here "X" is either "L" or "R" representing the left and right hemispheres respectively. "YY" can be FL, SU, LC, PL, TL or OL. These represent the frontal lobe, insula, limbic lobe, parietal lobe, temporal lobe and occipital lobe respectively. All other ROIs with the same colour code belongs to the same lobe and hemisphere as the one with text described here.

References

Satopaa, V. *et al.* (2011). Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171.